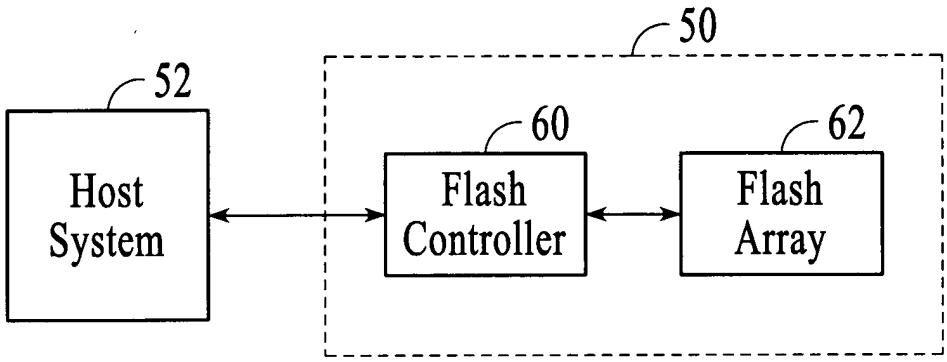


FIG.1
(PRIOR ART)

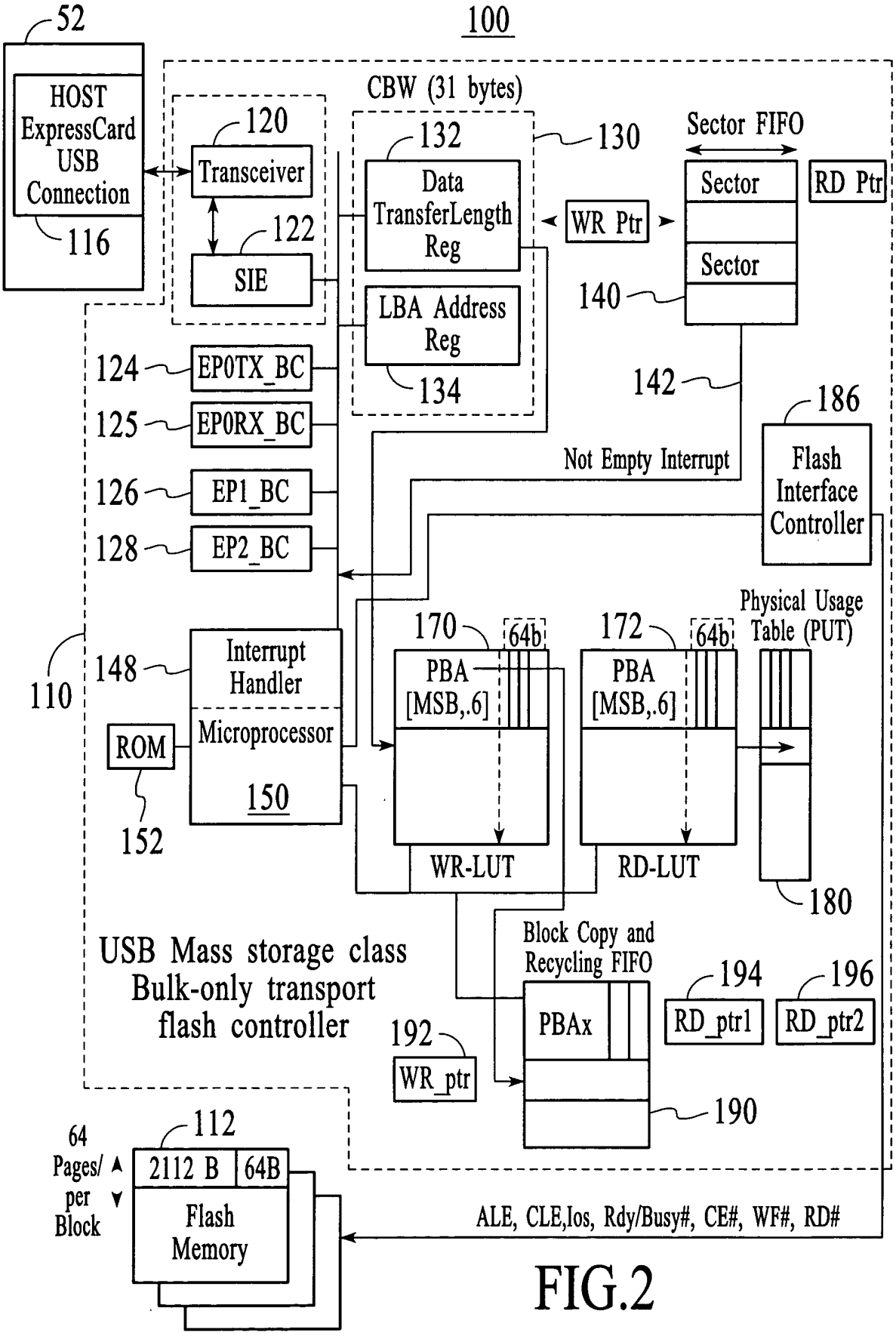

52
100
HOST ExpressCard USB Connection
116
CBW (31 bytes)
120
Transceiver
122
SIE
132
Data TransferLength Reg
130
Sector FIFO
Sector
RD Ptr
WR Ptr
Sector
140
LBA Address Reg
134
124
EP0TX_BC
125
EP0RX_BC
126
EP1_BC
128
EP2_BC
142
186
Not Empty Interrupt
Flash Interface Controller
Physical Usage Table (PUT)
110
148
Interrupt Handler
Microprocessor
150
ROM
152
170
64b
PBA [MSB,.6]
172
64b
PBA [MSB,.6]
WR-LUT
RD-LUT
180
USB Mass storage class Bulk-only transport flash controller
Block Copy and Recycling FIFO
194
196
RD_ptr1
RD_ptr2
192
WR_ptr
PBAx
190
64 Pages/ per Block
112
2112 B
64B
Flash Memory
ALE, CLE,Ios, Rdy/Busy#, CE#, WF#, RD#

FIG.2

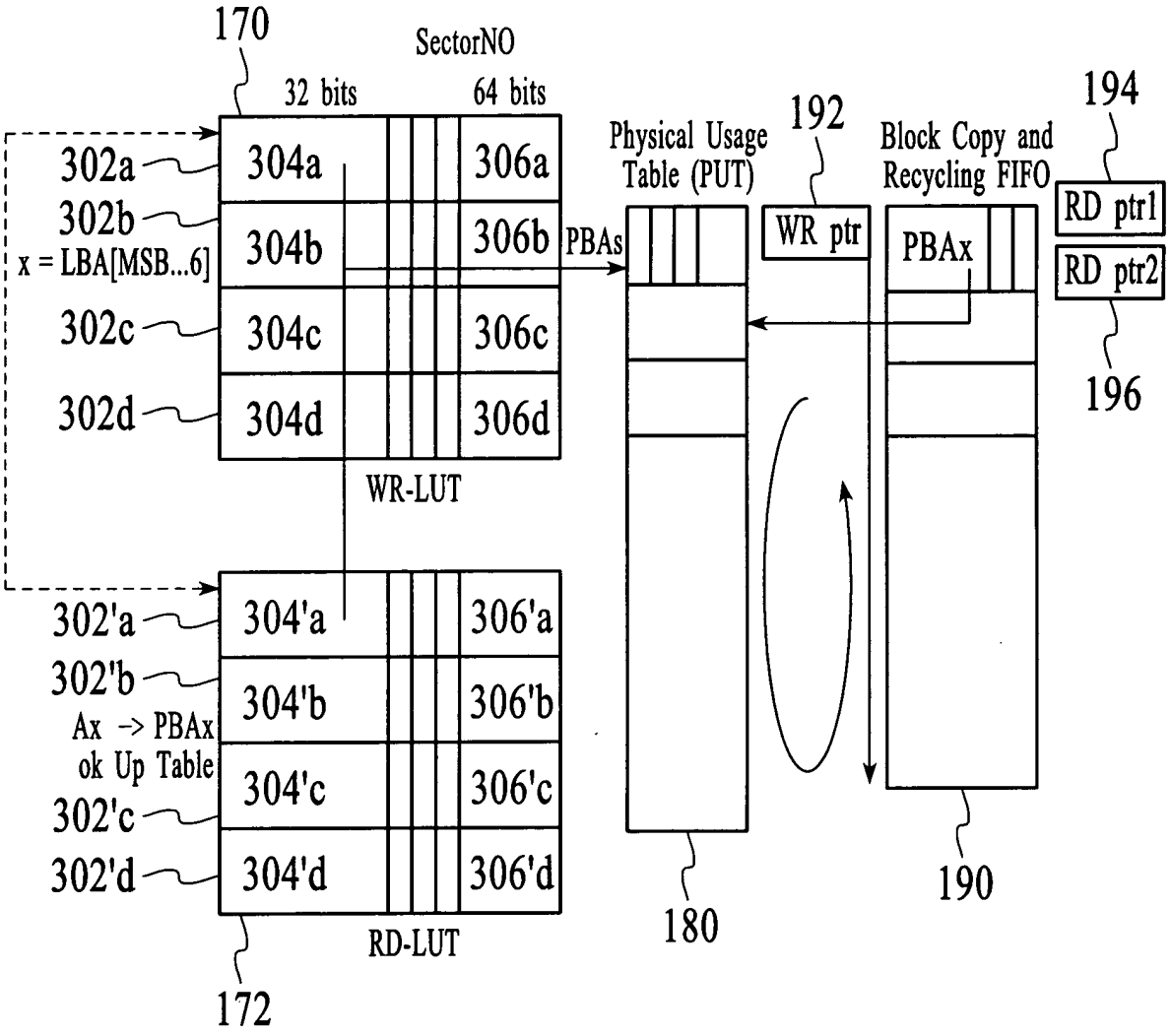

170
SectorNO
32 bits
64 bits
302a
304a
306a
302b
304b
306b
x = LBA[MSB...6]
302c
304c
306c
302d
304d
306d
WR-LUT
302'a
304'a
306'a
302'b
304'b
306'b
Ax -> PBAx
ok Up Table
302'c
304'c
306'c
302'd
304'd
306'd
RD-LUT
172
PBAs
Physical Usage
Table (PUT)
192
WR ptr
Block Copy and
Recycling FIFO
194
PBAx
RD ptr1
RD ptr2
196
180
190

FIG.3

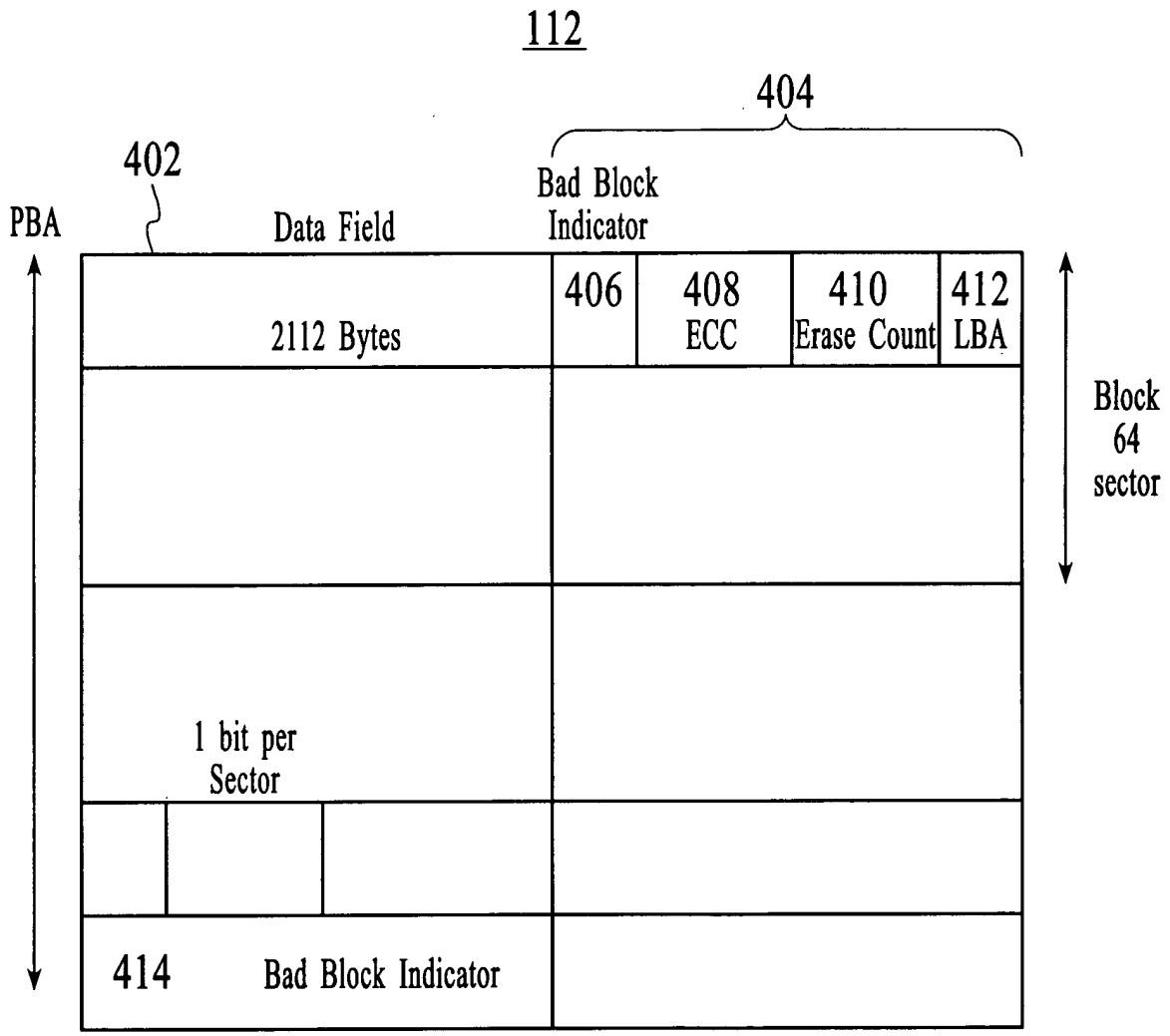

112
404
402
PBA
Data Field
Bad Block Indicator
406
408
ECC
410
Erase Count
412
LBA
2112 Bytes
Block 64 sector
1 bit per Sector
414
Bad Block Indicator

FIG.4

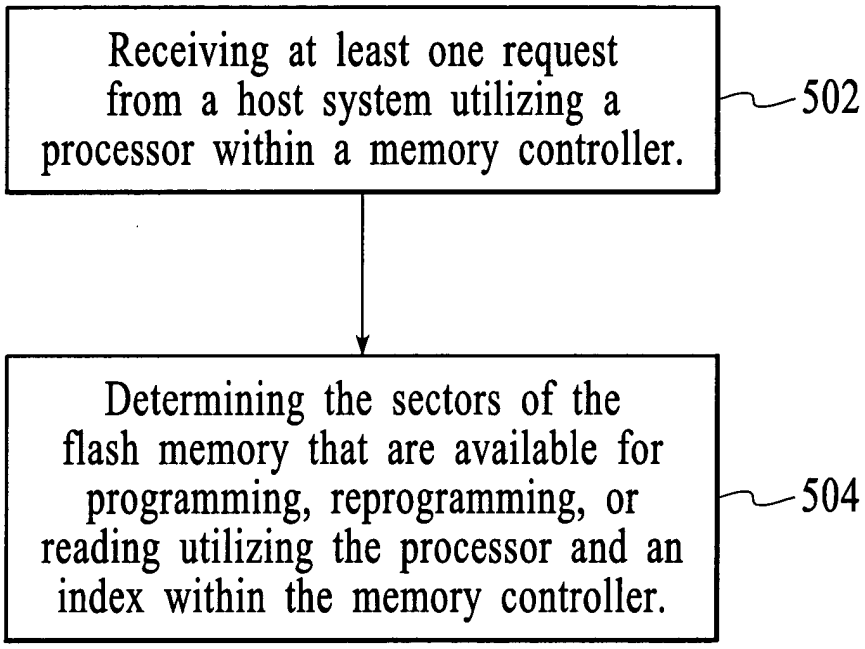

Receiving at least one request from a host system utilizing a processor within a memory controller.
502
Determining the sectors of the flash memory that are available for programming, reprogramming, or reading utilizing the processor and an index within the memory controller.
504

FIG.5

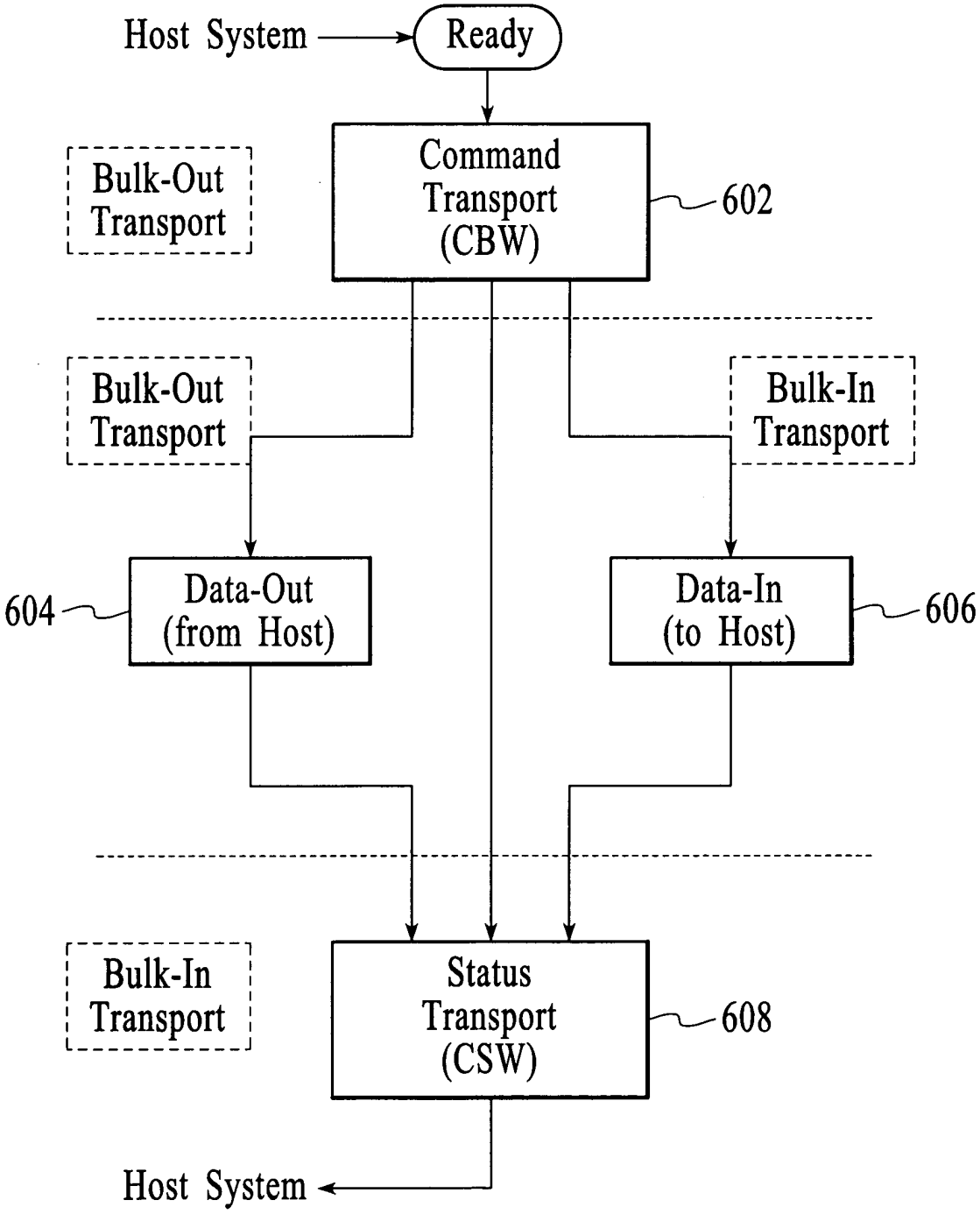

Host System
Ready
Command Transport (CBW)
602
Bulk-Out Transport
Bulk-Out Transport
Bulk-In Transport
604
Data-Out (from Host)
Data-In (to Host)
606
Bulk-In Transport
Status Transport (CSW)
608
Host System

FIG.6

702

Byte \ bit	7	6	5	4	3	2	1	0
0 - 3	CBW Signature (43 42 53 55h, 55 is byte 0)							
4 - 7	CBW Tag							
8 - 11 (O8h - OBh)	CBW Data Transfer Length							
12 (OCh)	CBW Flags							
13 (ODh)	Oh (Reserved)					CBW LUN		
14 (OEh)	Oh (Reserved)				CBW CB Length			
15 - 30 (OFh - 1Eh)	CBW CB							

FIG.7A

704

Byte \ bit	7	6	5	4	3	2	1	0
0	Operation Code (28h as read example)							
1	Reserved				Service Action			
2	MSB							
3								
4	LBA							
5	(Logical Block Address) LSB							
6	Reserved							
7	MSB	Transfer Length (unit is sector)						
8								LSB
9	Control (00h)							

FIG.7B

706

Byte \ bit	7	6	5	4	3	2	1	0
0 - 3	CSW Signature (53 42 53 55h, 55h is byte 0)							
4 - 7	CSW Tag							
8 - 11 (O8h - OBh)	CSW Data Transfer Length							
12 (OCh)	CSW Flags							

FIG.7C

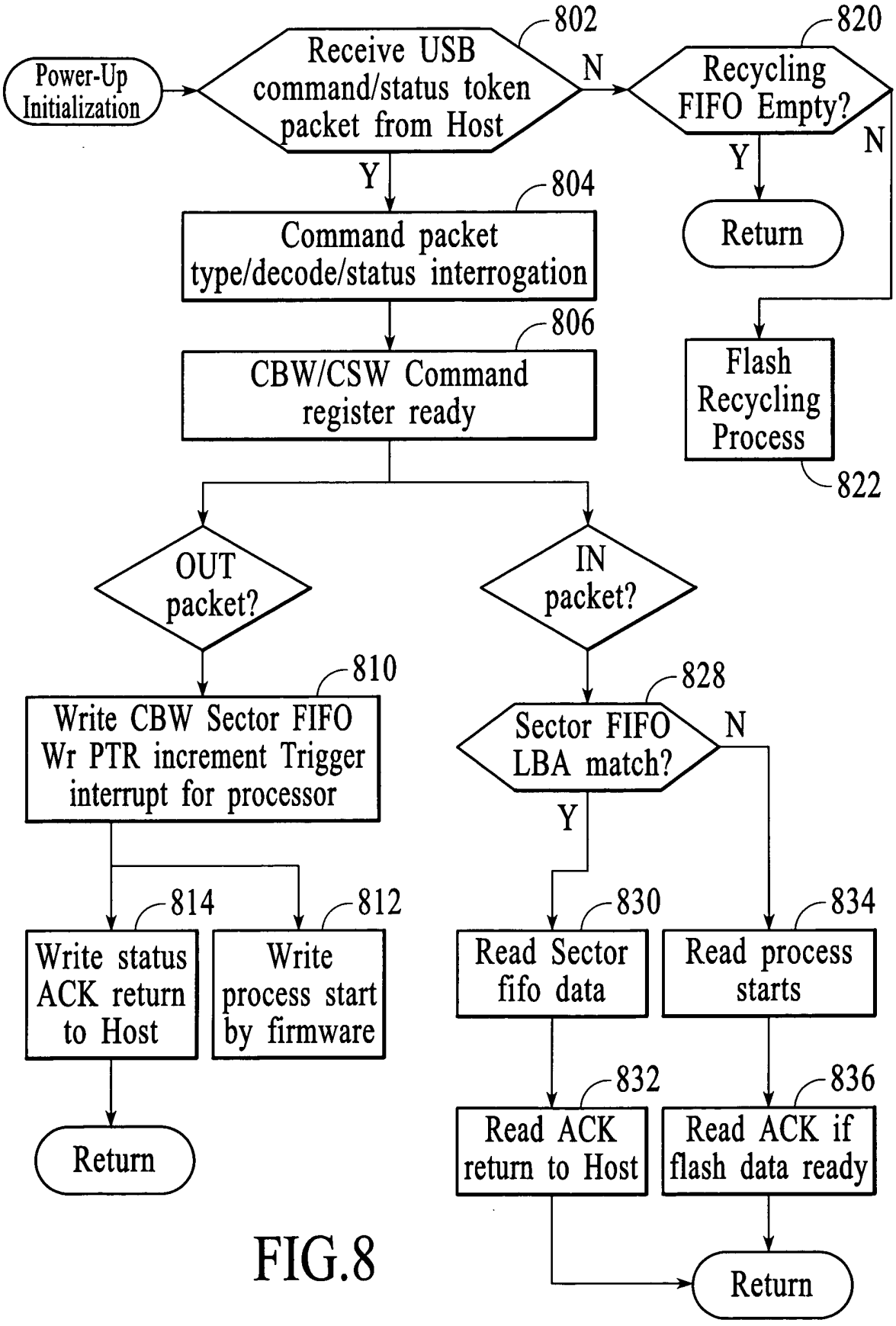

Power-Up Initialization
Receive USB command/status token packet from Host
802
N
Recycling FIFO Empty?
820
Y
N
Return
Y
804
Command packet type/decode/status interrogation
806
CBW/CSW Command register ready
Flash Recycling Process
822
OUT packet?
IN packet?
810
Write CBW Sector FIFO Wr PTR increment Trigger interrupt for processor
828
Sector FIFO LBA match?
N
Y
814
812
830
834
Write status ACK return to Host
Write process start by firmware
Read Sector fifo data
Read process starts
832
836
Return
Read ACK return to Host
Read ACK if flash data ready
Return

FIG.8

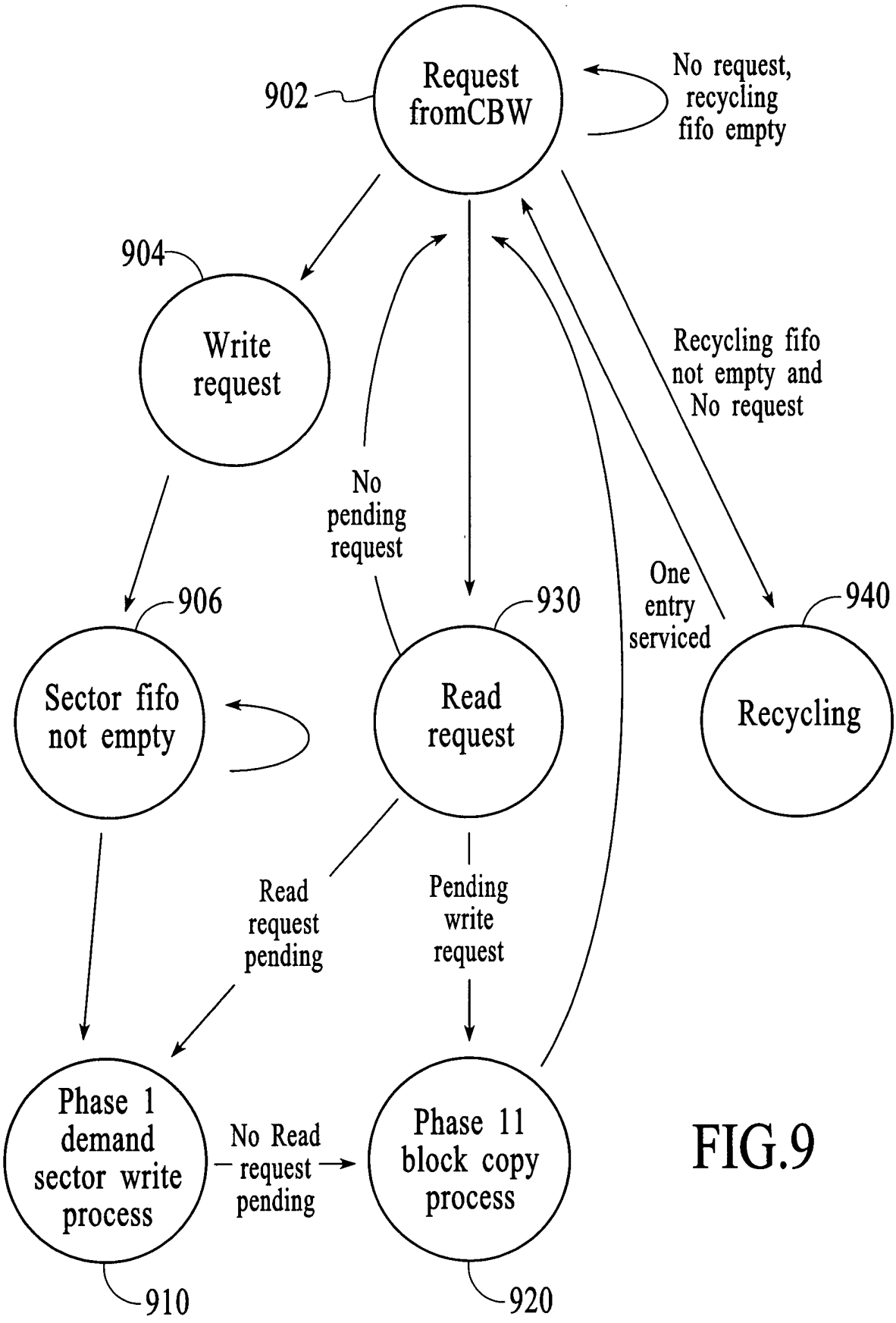

902
Request fromCBW
No request, recycling fifo empty
904
Write request
Recycling fifo not empty and No request
No pending request
One entry serviced
906
930
940
Sector fifo not empty
Read request
Recycling
Read request pending
Pending write request
Phase 1 demand sector write process
No Read request pending
Phase 11 block copy process
910
920

FIG.9

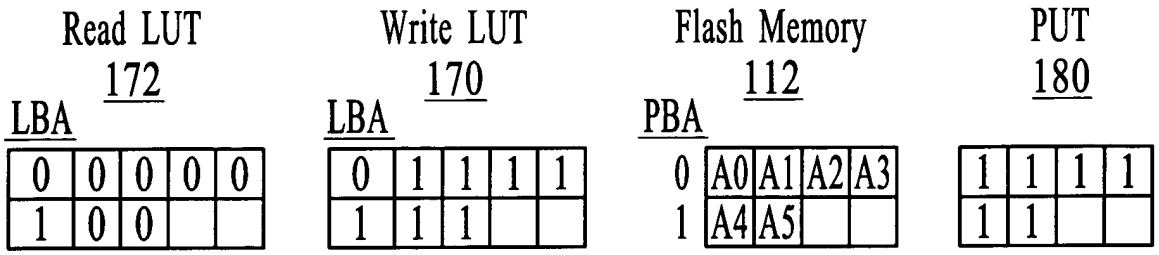


FIG.10A

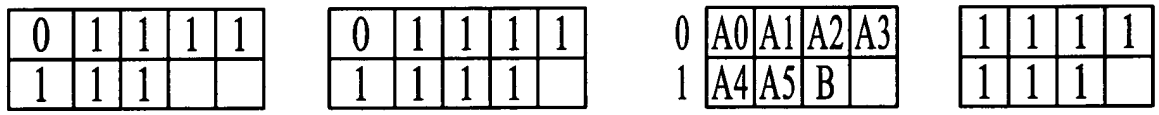


FIG.10B

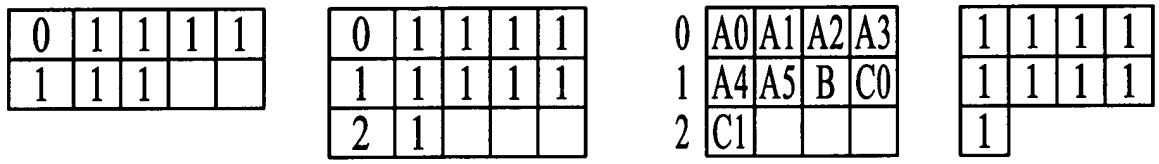


FIG.10C

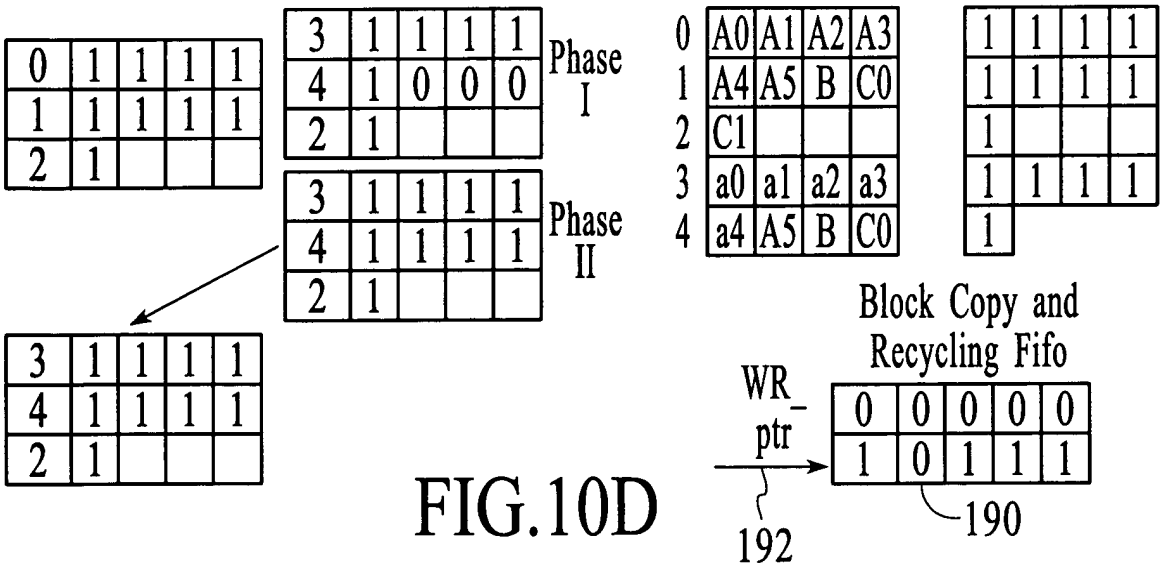
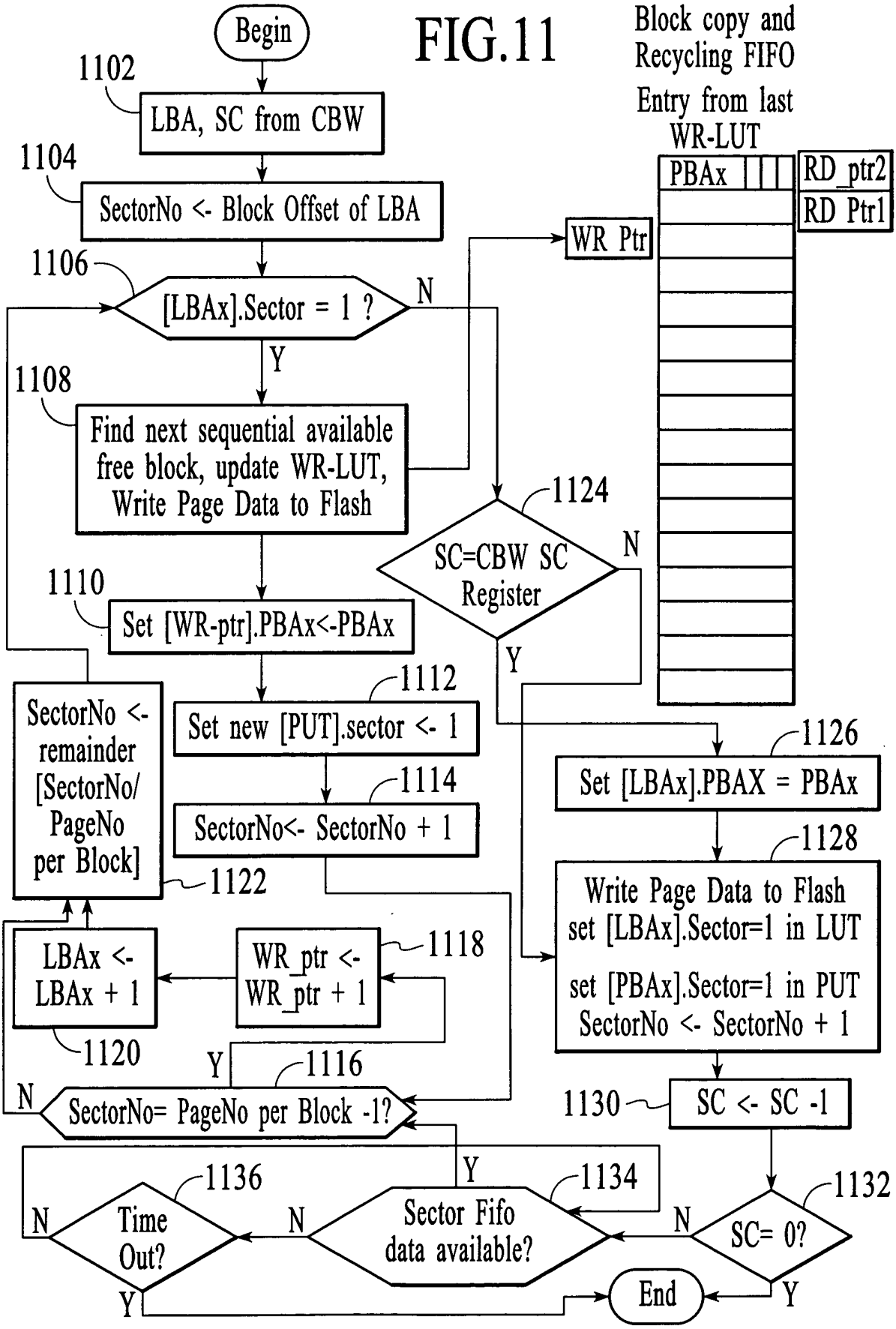


FIG.10D

FIG.11


Begin
1102
LBA, SC from CBW
1104
SectorNo <- Block Offset of LBA
1106
[LBAx].Sector = 1 ?
N
Y
1108
Find next sequential available free block, update WR-LUT, Write Page Data to Flash
1110
Set [WR-ptr].PBAx<-PBAx
1112
Set new [PUT].sector <- 1
1114
SectorNo<- SectorNo + 1
1122
SectorNo <- remainder [SectorNo/ PageNo per Block]
LBAx <- LBAx + 1
1120
WR_ptr <- WR_ptr + 1
1118
Y
1116
N
SectorNo= PageNo per Block -1?
Y
1136
N
Time Out?
Y
N
1134
Sector Fifo data available?
Y
N
1132
SC= 0?
Y
End
WR Ptr
1124
SC=CBW SC Register
N
Y
1126
Set [LBAx].PBAX = PBAx
1128
Write Page Data to Flash
set [LBAx].Sector=1 in LUT
set [PBAx].Sector=1 in PUT
SectorNo <- SectorNo + 1
1130
SC <- SC -1
Block copy and Recycling FIFO
Entry from last WR-LUT
PBAx
RD_ptr2
RD Ptr1

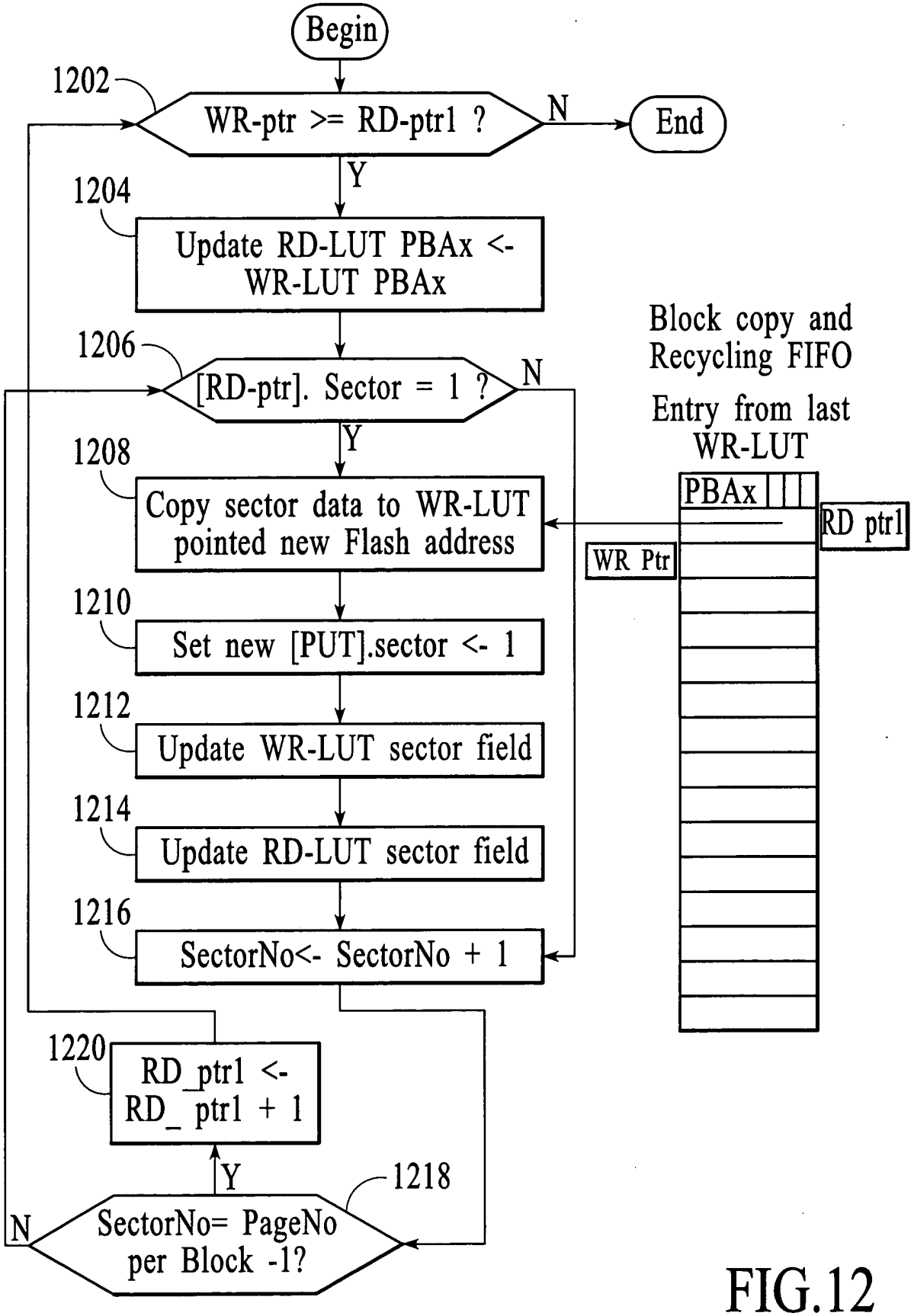

Begin
1202
WR-ptr >= RD-ptr1 ?
N
End
Y
1204
Update RD-LUT PBAx <- WR-LUT PBAx
1206
[RD-ptr]. Sector = 1 ?
N
Y
1208
Copy sector data to WR-LUT pointed new Flash address
1210
Set new [PUT].sector <- 1
1212
Update WR-LUT sector field
1214
Update RD-LUT sector field
1216
SectorNo<- SectorNo + 1
1220
RD_ptr1 <- RD_ ptr1 + 1
Y
1218
N
SectorNo= PageNo per Block -1?
Block copy and Recycling FIFO
Entry from last WR-LUT
PBAx
RD ptr1
WR Ptr

FIG.12

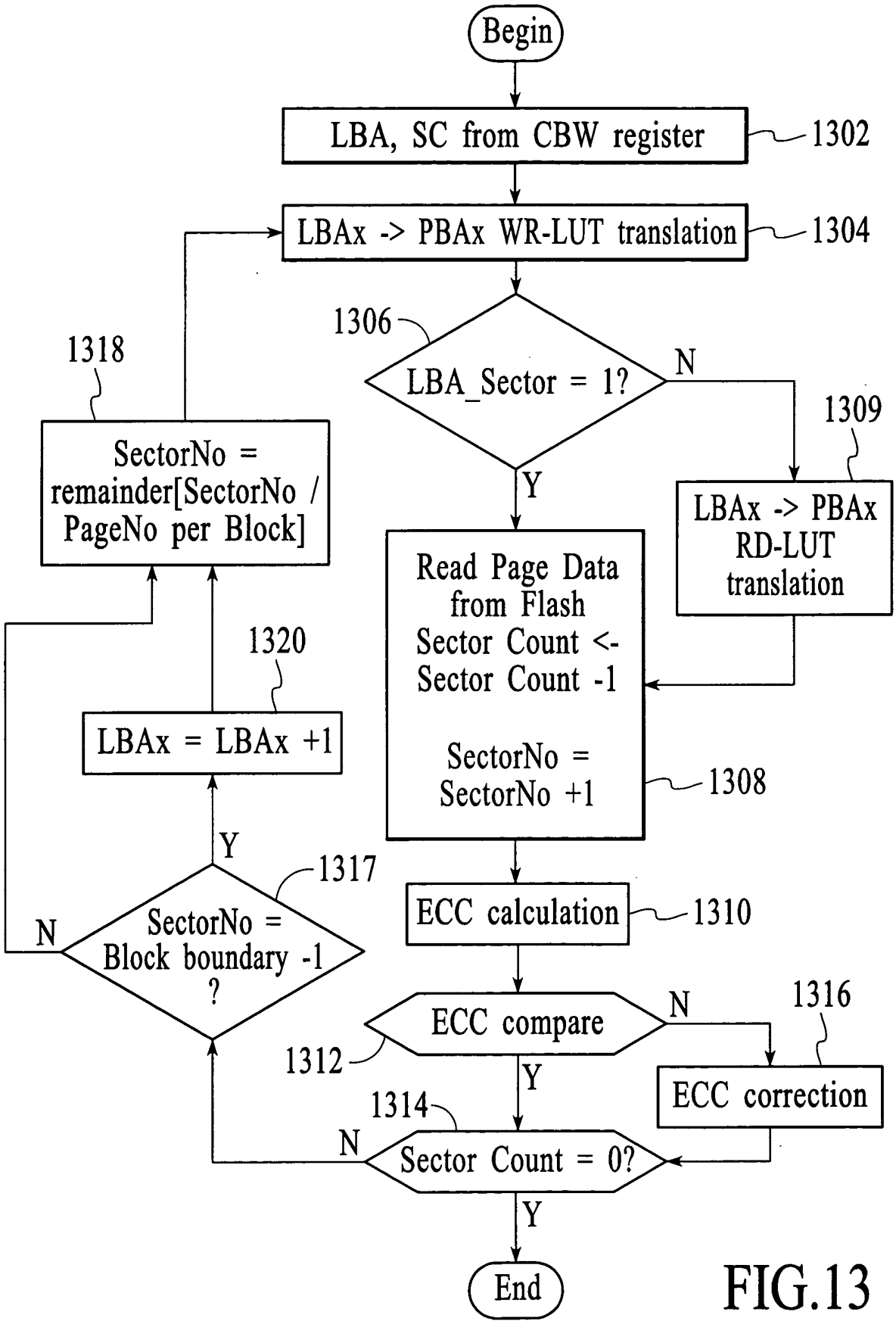

Begin
LBA, SC from CBW register
1302
LBAx -> PBAx WR-LUT translation
1304
1306
LBA_Sector = 1?
N
Y
1309
LBAx -> PBAx RD-LUT translation
Read Page Data from Flash
Sector Count <- Sector Count -1
SectorNo = SectorNo +1
1308
ECC calculation
1310
ECC compare
N
1316
ECC correction
1312
Y
1314
N
Sector Count = 0?
Y
End
1318
SectorNo = remainder[SectorNo / PageNo per Block]
1320
LBAx = LBAx +1
Y
1317
N
SectorNo = Block boundary -1 ?

FIG.13

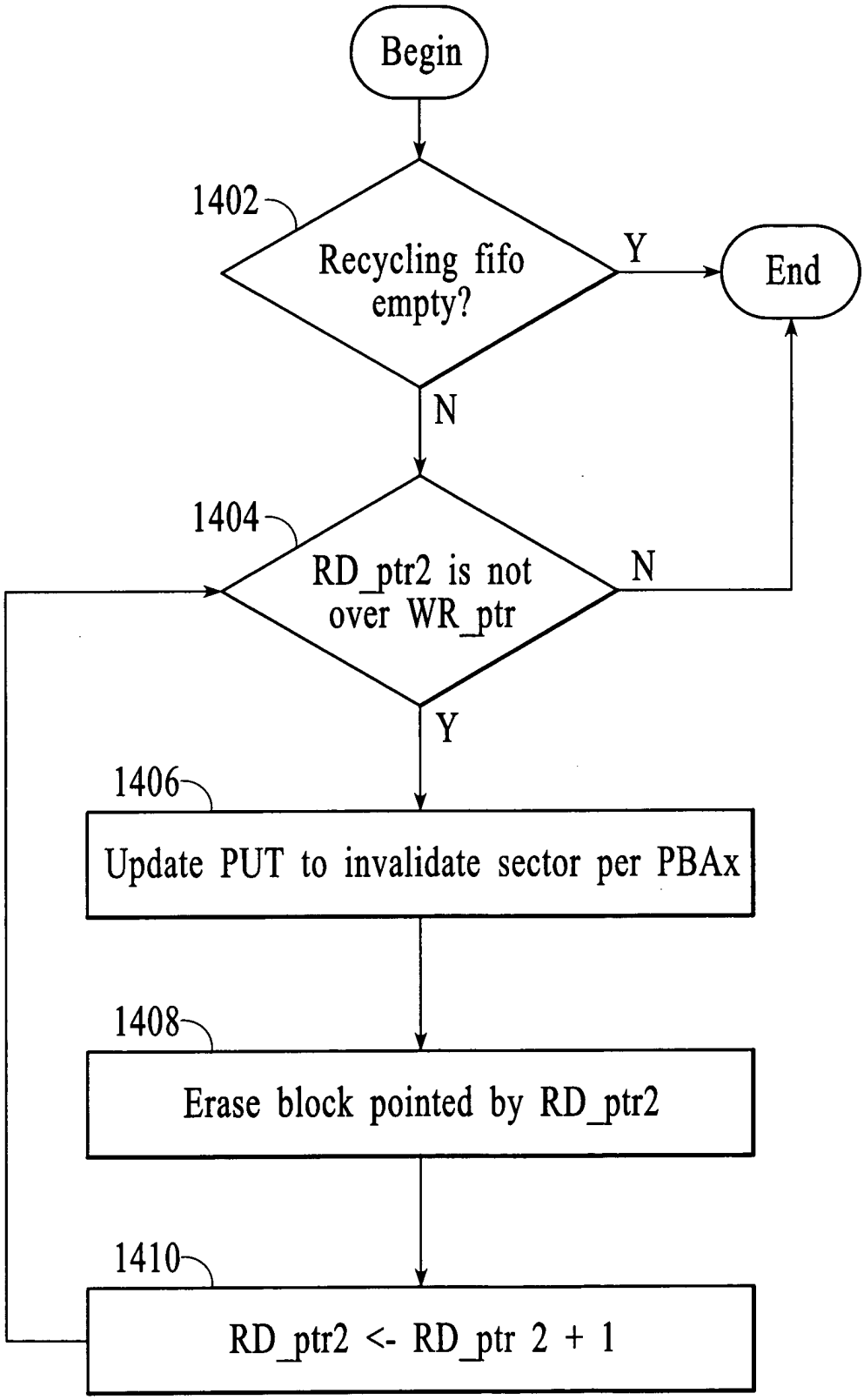

Begin
1402
Recycling fifo empty?
Y
End
N
1404
RD_ptr2 is not over WR_ptr
N
Y
1406
Update PUT to invalidate sector per PBAx
1408
Erase block pointed by RD_ptr2
1410
RD_ptr2 <- RD_ptr 2 + 1

FIG.14